International Union for Conservation of Nature Ecosystem Visualizer Zhuoming Tan '16, Son Le '16 (Sponsor: Professor John Magee, Professor Florencia Sangermano)

Introduction

We present an interdisciplinary research project as a visualizer that will help conservation scientists model and simulate ecosystems. It started as a request from the geography department of Clark University and is now being developed with close collaboration between computer science students and advisers in computer science and geography departments. To build a tool for scientists and field surveyers to facilitate the visualizing process of their sketchs of ecosystems, we utilize Java framwork JGraphX. It is capable of drawing and exporting biological system diagrams, whose specifications come from Internation Union for Conservation of Nature, to have a standardized way of modeling the system so that the publications and presentations share a common presentation approach.

Why Use a Visualizer?

Identification of key ecological processes is an essential element of the ecosystem description. Therefore, geologists and environmental scientists use diagrams in their reserach papers and presentations to represent their conceptual models they create from their research and field surveys. An example is the Fig. 1 below.

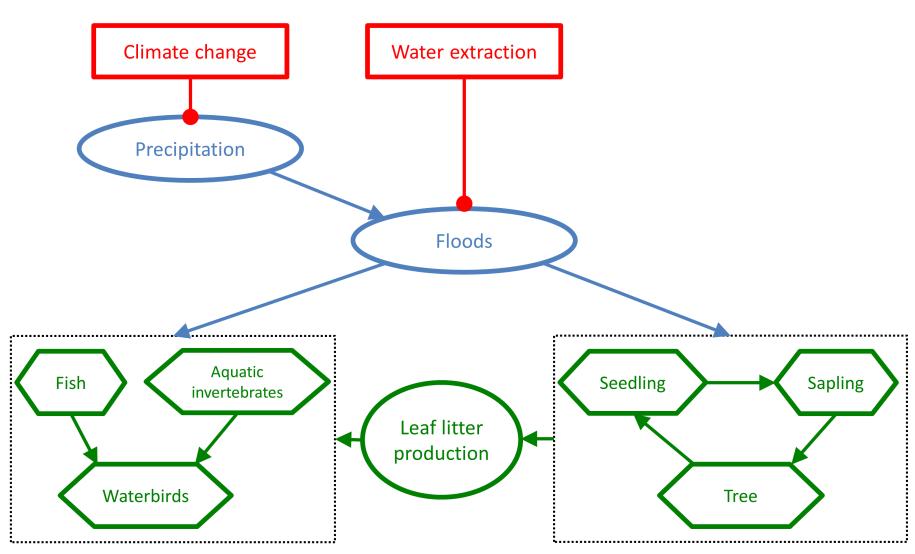


Figure 1: Cause-effect model of the Gonakier forest in Senegal [1]

The scientists look for a way to streamline and standardize the process, so that these conceptual diagrams look more uniform, are easier to create, modify, and exchanged. Furthermore, there is demand in simple analyzing power in the software, capable of determining the effect of some of the factors.

Computer Science Department, Clark University | Worcester, MA

Choice of Framwork—JGraphX

Licensed under BSD license, the JGraphX framework is a Java framework that could be used to build a desktop applet enabled to create and process graphs, which could be easily saved and opened, visualized, analyzed, and interacted with.

It has the possibility to be customized. The shapes and relationship we need for IUCN guidelines could be coded into the framework, and lightweighted database and analysis functionalities are possible to be integrated.

As a familiar language for both of the developers, Java is the language of choice when it comes to faster adaptation, and less learning overhead.

Lastly, one of the requirements for the app is to be cross-platform. This choice of language and framework suits this purpose.

Important Result

This program could now produce conceptual model diagrams that conform to the IUCN recommendations. Different shapes, arrowheads, and compartment are all implemented.

Implementation

To customize our editor GraphEditor, we have to modify a sample program provided by JGraphX, and extend that class **BasicGraphEditor**.

We also implemented our own version of DecoratedConnectorShape which extends mxConnectorShape, and ParallelogramShape which extends **mxBasicShape**.

Other different shapes are modified based on the basic shapes provided, using XML. One of the difficulties is to correctly define the perimeters of the shapes, to perfectly attach the arrows to the shapes.

The UI buttons of these shapes are drawn in vector drawing program and attached to the program. However, we have yet to figure out proper way to do antialiasing.

Currently, the project is hosted on a private GitHub repository. Basic functionalities are in place, but we look forward to further perfection of the documentation, and improvement and implementation on a few requested functionalities, most notably the database and the analysis ability. The program prototype has been distributed to end-users, and constructive feedbacks have been collected since. They could direct the path of future work.

Screen Shot

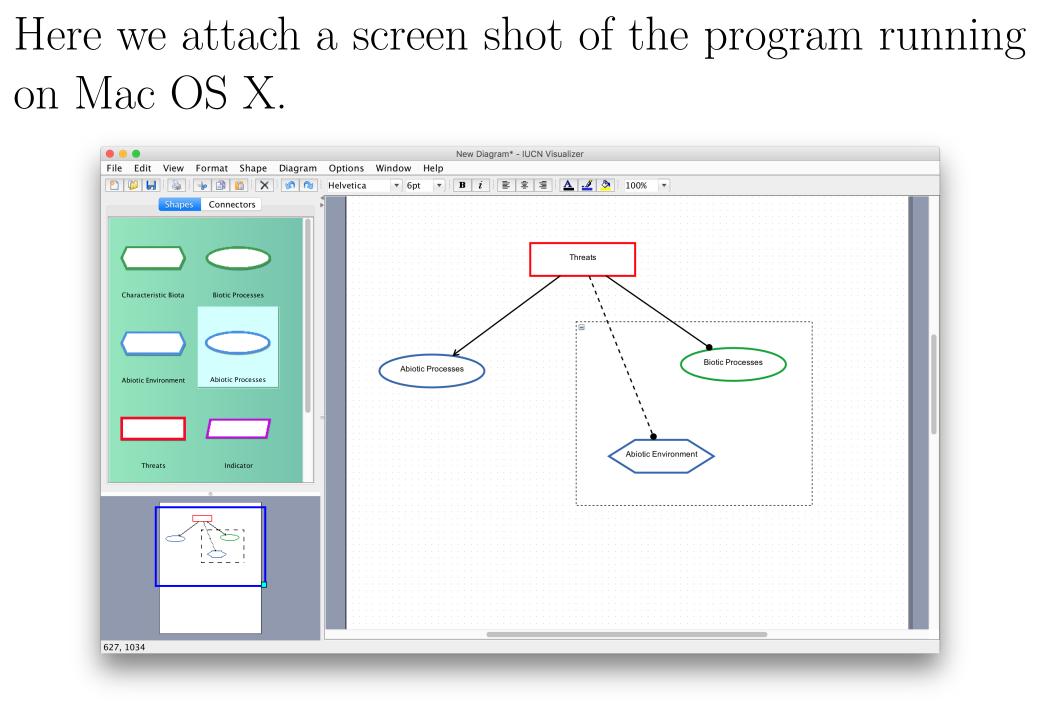


Figure 2: Screen Shot of the Program

Development Status

	📓 GraphEditor.java - IUCN - [~/Documents/IUCN]	
IUCN rc src rc main rc jav	ra > 🗈 edu > 🗈 clarku > 🗈 IUCN > 🗈 view > 逮 GraphEditor >	GraphEditor 🔻 🕨 🎉 🥰 🖓 🖕 🖬 🔍
मु 🗊 Project• 😯 崇 🌞 📂	© DecoratedConnectorShape.java × C GraphEditor.java × C ParallelogramShape.java × C mxGraph.java ×	m
Image: Project Imag	<pre>1 package edu.clarku.IUCN.view; 2 3 import 37 37 38 public class GraphEditor extends BasicGraphEditor { 39 /** 40 * 41 */ 42 private static final long serialVersionUID = -4601740824088314699L; 43</pre>	Maven Projects
 demo models shapes DecoratedCo arallelogram tuil view editor images resources 	<pre>44 /** 45 * Holds the shared number formatter. 46 * 47 * @see NumberFormat#getInstance() 48 */ 49 public static final NumberFormat = NumberFormat.getInstance(); 50 51 /** 52 * Holds the URL for the icon to be used as a handle for creating new 53 * connections. This is currently unused. 54 */ 55 * public GraphEditor() { this("IUCN Visualizer", new CustomGraphComponent(new CustomGraph())); } 58 59 * /**</pre>	Jase
itignore ravis.yml JCN.iml om.xml EADME.md nal Libraries	<pre>60 ** 61 */ 62 #* 62 #* 63 super(appTitle, component); 63 final mxGraph graph = graphComponent.getGraph(); 64 final mxGraph graph = graphComponent.getGraph(); 65 // Creates the shapes palette 67 EditorPalette shapesPalette = insertPalette(mxResources.get("shapes")); 68 EditorPalette connectorPalette = insertPalette(mxResources.get("connectors")); 69 // Sets the edge template to be used for creating new edges if an edge 71 // is clicked in the shape palette 72 shapesPalette.addListener(mxEvent.SELECT, (sender, evt) -> { 73 Object tmp = evt.getProperty("transferable"); 74 if (tmp instanceof mxGraphTransferable) { 75 mxGraphTransferable t = (mxGraphTransferable) tmp; 76 Object cell = t.getCells()[0]; 78 if (araph.getModel().isEdge(cell)) { 78 } 79 } 79 } 70 } 71 } 72 } 73 } 73 } 74 } 75 }</pre>	
	: TODO 🛛 🔲 Event Log 🛛 🧕 9: Version Control	
All files are up-to-date (20 minut	es ago)	60:7 LF¢ UTF-8¢ Git: develop¢ 🚡 👮

The project successfully addresses a real word demand for a convenient visualizing tool for biosystem conceptual models. This tool is offline, cross-platform, and portable. The program enables easy drawing, saving, and exchange of such diagrams, and convenient export to most format people work with, including vector format which is suitable for publishing. The project has also left great room for improvement. At current stage the software is capable of drawing diagrams via drag and drop from predefined gadgets list. Future planned functions include simple analyzing ability, which qualitatively indicates the effect the change of the factors will have on the biosystem. A prototype version has been shared with IUCN scientists and the first round of feedback is being incorporated into ongoing development.



[1] David A Keith, Jon Paul Rodríguez, Kathryn M Rodríguez-Clark, Emily Nicholson, Kaisu Aapala, Alfonso Alonso, Marianne Asmussen, Steven Bachman, Alberto Basset, Edmund G Barrow, et al. Scientific foundations for an IUCN Red List of Ecosystems. *PLOS one*, 8(5):e62111, 2013.

Thank Professor Florencia Sangermano of Geography Department, Clark University for being the bridge between the users and the developers. Thank Professor John Magee for providing guidance in software engineering practices.

Figure 3: Screen Shot of the Development Progress

Conclusion

	າງ Pull requests o 📰 Wiki ≁~ Pulse	III Graphs 🔅 Settings		
IUCN Visualizer — Edit	۶۶ 4 branches	♡ O releases	$rac{1}{\sqrt{2}}$ 1 contributor	
Branch: develop - New pull r	equest New file Upload files Find f	HTTPS → https://github.com	n/tzm41/ 🔂 😫 Download ZIP	
This branch is 52 commits ahead o	of master.	🏌 Pull request 🗈 Compare		
tzm41 Merge pull request #7 f	rom tzm41/menuBar		Latest commit 79c2825 on Feb 19	
src	Updated to support word wrapping		2 months ago	
juitignore	Clean up directory		4 months ago	
.travis.yml	Restructure into a Maven project.		5 months ago	
README.md	Restructure into a Maven project.		5 months ago	
Dom.xml	Correct path to project's resources.		4 months ago	
E README.md				
IUCN				

Figure 4: Project page on GitHub

References

Acknowledgements

